Bayesian Learning and Montecarlo Simulation, Finite Mixture Models on Lake Acidity Dataset

Davide Villani, Emanuele Caruso

Abstract

In the following report we present a clustering method based on Finite Mixture Models applied to the *Acidity* dataset which shows the log acidity index for 155 lakes in the Northeastern United States. In particular we provide an estimate of the total density and of the clustering of data with Mixture Normals using different priors and compare how these changes effect the final results and the posteriors.

1 Introduction

1.1 Dataset Description

The dataset used in this study is taken from the R package '*gamlss.data*' and describes the acidity index for 155 lakes in the Northeastern United States.

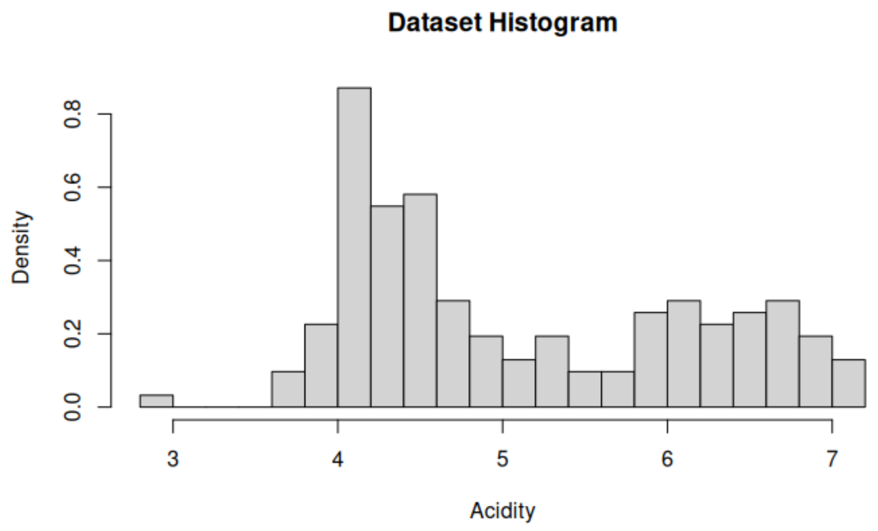


Figure 1: Probability Histogram to show how data is spread based on their acidity index.

1.2 Dataset Exploration

Before getting into the Bayesian implementation of the clustering we performed a standard *K-means* clustering and computed a plot to show the *Variability Ratio* using a number of clusters varying from 1 to 10. Even if this is out of the scope of the project this will help us understand how many possible clusters are present in the dataset without choosing completely at random.
In particular the Variability Ratio describes for each k-cluster the ratio between the *Within Cluster Sum of Squares* of each cluster and the total squared sum, which is just *WSS* + *Between Cluster Sums of Squares* , what we generally want to do is to minimize the WSS while maximizing the BSS.

$$\text{WSS} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \qquad \text{BSS} = \sum_{k=1}^{K} n_k \|\mu_k - \mu\|^2$$

where μ_k are the centroids for each cluster, μ is the overall mean, x_i are the single observations and n_k are the number of observations in each cluster k.

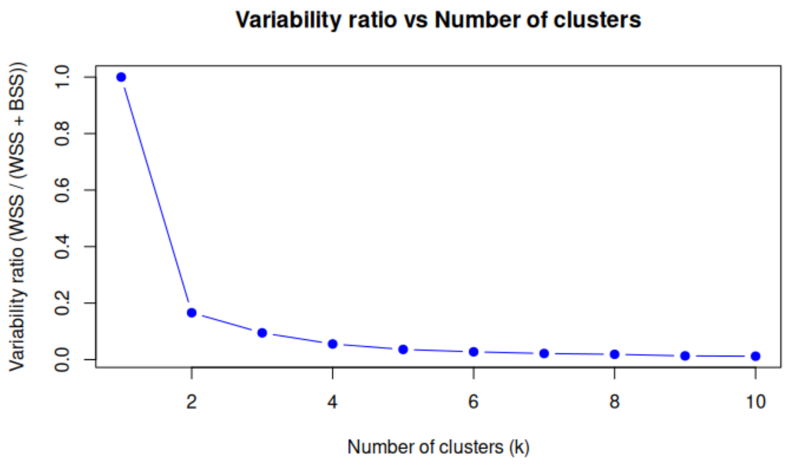
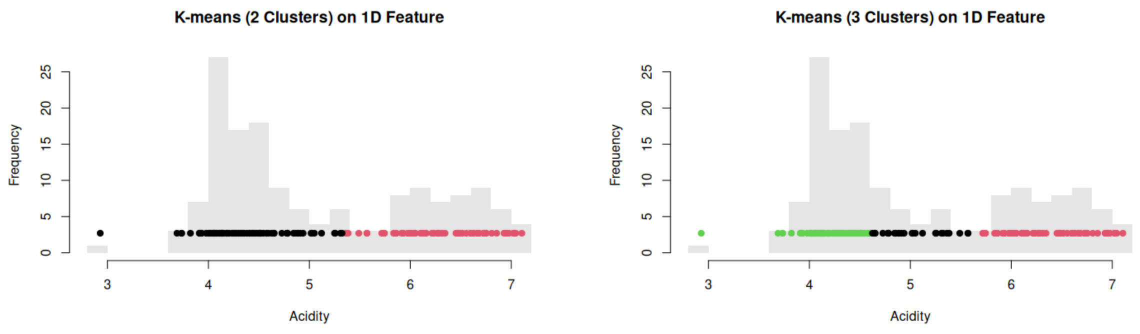


Figure 2: Variability Ratio using K-Means

As the plot suggest, by using the *Elbow Method* we can see how a good number of possible clusters is 2, in Fig. 3 we can compare two clustering results using k = 2,3



(a) K-Means for k equal 2 (b) K-Means for k equal 3

Figure 3: K-Means

2 Gaussian Mixture Model

2.1 Model Specification

The density of an observation y_t in a finite mixture model is

$$f(y_t) = \sum_{h=1}^{K} w_h \, K(y_t \mid \theta_h) \quad , \quad \sum_{h=1}^{K} w_h = 1.$$

where $K(y \mid \theta)$ is a kernel density defined over the product of the observation space $\mathcal{Y}$ and the parameter space Θ.
In our study we decided to use a Normal distribution as our Kernel, this means that our data is spread as a combination of K Normals and, most importantly, the sum of the weights w_h of each component needs to be equal to 1. This is fundamental because the weights represents the probability that a randomly chosen data point was generated by the $k-th$ Gaussian component and since the data must come from one of the components, the total probability must sum to 1.
In our case we choose the kernel density as:

$$K(y \mid \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right), \qquad \theta = (\mu, \sigma) \in \Theta = \mathbb{R} \times \mathbb{R}^+, \quad \mathcal{Y} = \mathbb{R}$$

2.2 Model Simulation

Since our total density is represented by multiple densities and we also need to define the probabilities of an observation falling into one of the k components, we can't just sample from the full density but we need to apply a different technique.
The main idea is to randomly choose with probability w_h the component's parameters $\theta_h = (\mu_h, \sigma_h)$ and only then sample y_t at random from our $h-th$ Gaussian component.
In this way we represent the mixture model as :

$$y_t \mid \vartheta_t \overset{\text{ind}}{\sim} K(y \mid \vartheta_t), \quad t = 1, \dots, T$$

$$\vartheta_t \mid G \overset{\text{ind}}{\sim} G(\vartheta) = \sum_{h=1}^{K} w_h \, \delta_{\theta_h}(\vartheta)$$

where δ_θ is the Dirac point mass at θ, a distribution that places all its probability at the single point θ. That is, the ϑ_t are latent variables such that they contains the information on what parameters we will use for sampling y_t.
Another way to see this, which is more suitable to be coded, is for each observation t, we introduce a variable $D_t \in \{1, \dots, T\}$ indicating which component of the mixture generated the observation.
At this point we can express the joint density of y_t and D_t given the mixture weights $\mathbf{w}$ and the component parameters $\theta = (\theta_1, \dots, \theta_K)$ as:

$$f(y_t, D_t \mid \mathbf{w}, \theta) = f(D_t \mid \mathbf{w}) \, f(y_t \mid D_t, \theta) = w_{D_t} \, K(y_t \mid \theta_{D_t}).$$

And by integrating out the latent variable D_t, we recover the standard finite mixture model density:

$$f(y_t \mid \mathbf{w}, \theta) = \sum_{h=1}^{K} f(y_t, D_t = h \mid \mathbf{w}, \theta) = \sum_{h=1}^{K} w_h \, K(y_t \mid \theta_h).$$

We can notice that there is an equivalence with the latent variable representation used earlier $\theta_{D_t} = \vartheta_t$.
Thanks to this new model we say that two samples y_t and y_s are in the same cluster if $D_t = D_s$, that is, if they share the same parameter $\vartheta_t = \vartheta_s$.

2.3 JAGS Settings

JAGS (*Just Another Gibbs Sampler*) is a framework to perform Gibbs sampling, a Markov Chain Monte Carlo (MCMC) method that takes advantage of cases where we have a target distribution described by $f(\theta_1, \theta_2|x)$ but for example it's not possible to directly sample from the posterior $f(\theta_2|x)$ but we can instead sample from $f(\theta_1|\theta_2, x)$ and $f(\theta_2|\theta_1, x)$ the Full Conditionals.
In our case of Gaussian Mixture Model let's consider $\theta_h = (\mu_h, \sigma_h^2)$ for each h component of the mixture, our priors in every model will be $\mu_h{\sim}\mathcal{N}(\mu_0, \sigma_0)$ and $\frac{1}{\sigma_h^2}{\sim}\text{Gamma}(\alpha_0, \beta_0)$.
Using a normal kernel will give us that:

$$\mu_h \mid \sigma_h, \mathbf{w}, D, y \sim \mathcal{N}(\mu_h^T, \sigma_h^T), \qquad \frac{1}{\sigma_h^2} \mid \mu_h, \mathbf{w}, D, y \sim \mathcal{G}(a_h^T, b_h^T)$$

where

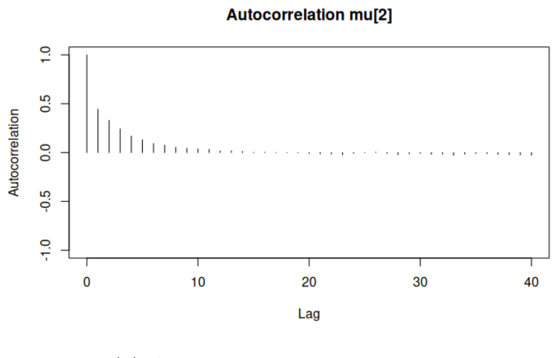
$$\mu_h^T = \frac{\sigma_h^2}{\sigma_h^2 + |D_h|\sigma_0^2}\mu_0 + \frac{|D_h|\sigma_0^2}{\sigma_h^2 + |D_h|, \sigma_0^2} \cdot \frac{1}{|D_h|}\sum_{t \in D_h} y_t, \qquad \sigma_h^T = \frac{\sigma_h^2\sigma_0^2}{\sigma_h^2 + |D_h|\sigma_0^2}$$

$$a_h^T = \alpha_0 + \frac{|D_h|}{2}, \qquad b_h^T = \beta_0 + \frac{1}{2}\sum_{t \in D_h}(y_t - \mu_h)^2$$

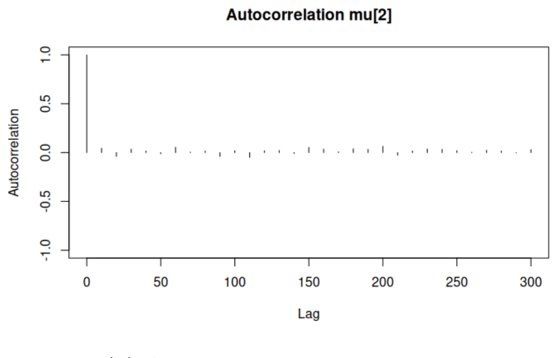
This is straightforward once we notice how for every component we have a Normal-Normal-Inverse Gamma model, which is a known conjugate model.
So we can now see how our full conditionals are exact known distributions which makes the *Gibbs* sampling the most suitable technique. For our model, we configured the MCMC sampler as follows:

- `num_chains = 3`: We use three independent Markov chains to assess convergence.
- `n.iter = 10000`: A total of 10,000 iterations are performed for each chain.
- `burn_in = 1000`: The first 1,000 samples from each chain are discarded to allow the sampler to reach the high-probability region of the posterior (i.e., to reduce initialization bias).
- `thin = 10`: To mitigate the effects of autocorrelation, only when the 10th sample is retained.

To show why thinning is particularly important here is an autocorrelation plot showing a sampling with and without thinning focusing on the randomly picked μ_2:



(a) Autocorrelation with thin = 1



(b) Autocorrelation with thin = 10

As expected with MCMC methods, autocorrelation is high at lag 1. However in the second plot, it rapidly decays to near zero at higher lags. This indicates that the thinning strategy—retaining one sample every 10 iterations—effectively reduces autocorrelation in the posterior samples.

2.4 Priors

In this section we compare the results of using different priors and how they affect the final density. We first need to define the distribution of the latent variable which we call z.

2.4.1 Bernoulli

Since our previous dataset explorations suggests our data is divided into 2 components, it is particularly convenient to use a Bernoulli density function to model our latent variable.

$$Y_1, \dots, Y_n \overset{\text{i.i.d.}}{\sim} w\,\mathcal{N}_1\left(\mu_1, \sigma_1^2\right) + (1-w)\,\mathcal{N}_1\left(\mu_2, \sigma_2^2\right)$$

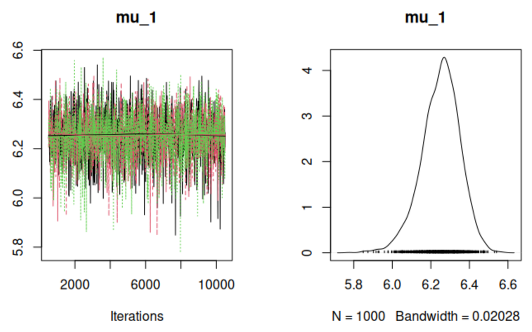
$$Z_1, \dots, Z_n \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(w)$$

$$w \sim \text{Beta}(2,2)$$

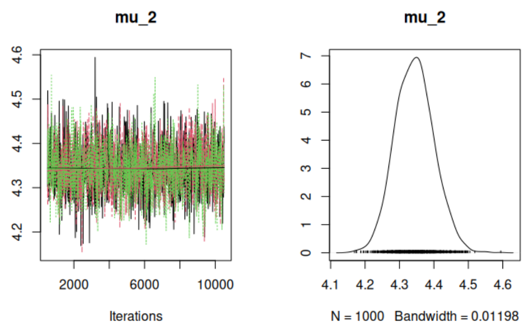
$$\begin{array}{l} \mu_1 \sim \mathcal{N}(\mu_{0,1},\, 1), \\ \mu_2 \sim \mathcal{N}(\mu_{0,2},\, 1), \end{array} \quad \text{where } \mu_{0,1} \neq \mu_{0,2}$$

$$\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2} \overset{\text{i.i.d.}}{\sim} \text{Gamma}(3,3)$$

In order to sample and compute the posteriors of this model we used *JAGS* and run 3 chains and we obtained the following posteriors for the means μ_1, μ_2 the variances σ_1^2, σ_2^2 and the Bernoulli weight w.

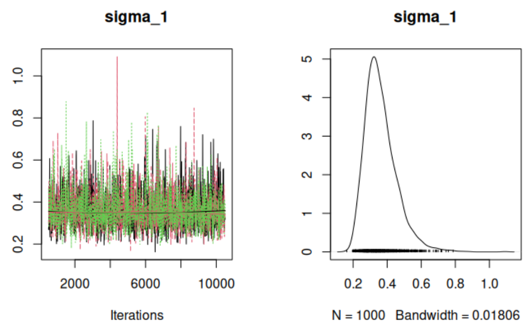


(a) μ_1

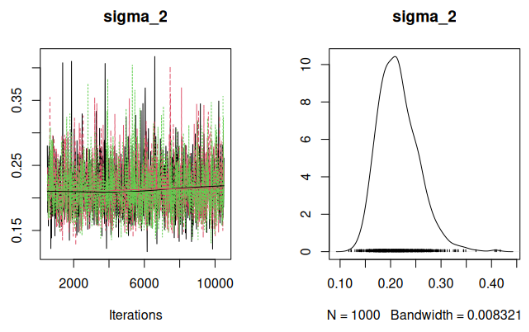


(b) μ_2

Figure 5: Traceplot and marginal posterior for the means

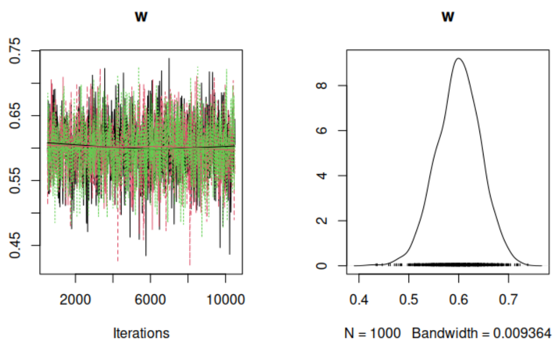


(a) σ_1^2

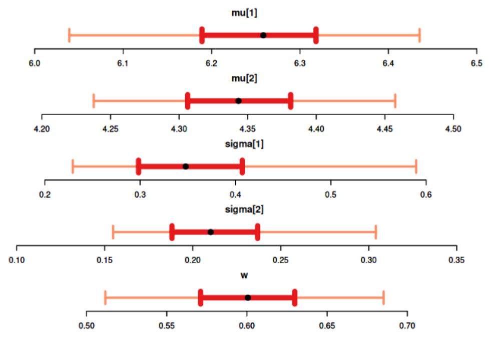


(b) σ_2^2

Figure 6: Traceplot and marginal posterior for the variances



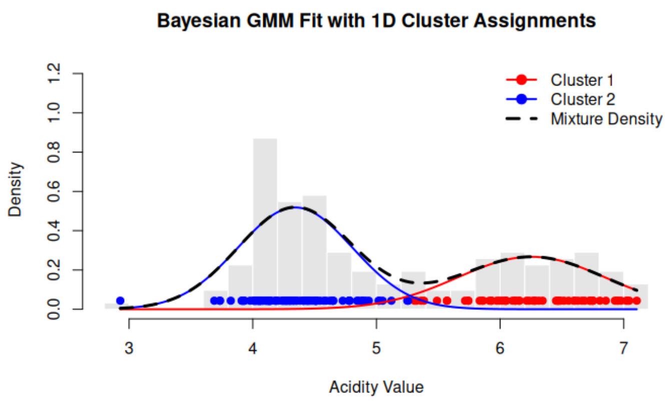
(a) w



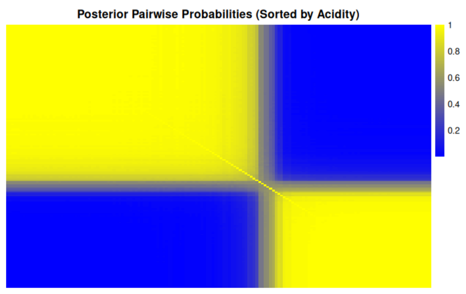
(b) Credible Intervals for each parameter, in red the 50% and in orange 95% CI

Figure 7: Traceplot and marginal posterior for the weight and CI plot

From Figs 5,6,7 we notice how the traceplots are behaving correctly remaining well-separated and stable throughout the sampling process, suggesting that the sampler has consistently assigned each parameter to the same underlying component across iterations. This indicates good mixing and convergence of the 3 MCMC chains (presented in the traceplots in 3 colors, Red Black and Green). Importantly, we do not observe any evidence of **label switching** common in mixture models where component labels are interchangeable, leading to traceplots that flip or cross over time. In the next section we will deal with more than 2 component's mixture and we will see how to identify label switching and how to avoid it.



(a) Final Plot of densities and cluster assignments



(b) Heatmap of the Posterior Pairwise Probabilities

Looking at the final assignments and the total mixture density, we can see how the two weighted densities actually describe the dataset coherently to what we expected by looking at the histogram. We then computed the Posterior Similarity Matrix (PSM) to visualize the uncertainty in cluster assignments across posterior samples from the model.
Given M posterior samples of latent cluster assignments where $\mathbf{z}^{(m)} = (z_1^{(m)}, \ldots, z_T^{(m)})$ represents the cluster assignments for T observations in the m-th chain, the PSM is a $T \times T$ matrix P defined as:

$$P_{ij} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}\left\{ z_i^{(m)} = z_j^{(m)} \right\}$$

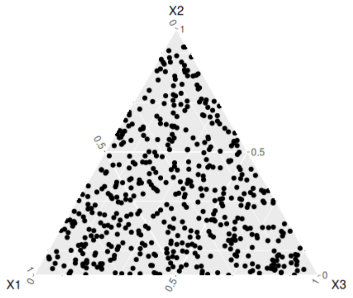
In Fig 11b we can see such matrix and notice the block structure with bright colored areas in all the matrix, revealing a low uncertainty which is only noticeable around the intersection between the blocks, in that area indeed the two components meet and assigning one observation to a cluster becomes more uncertain.

2.4.2 Categorical

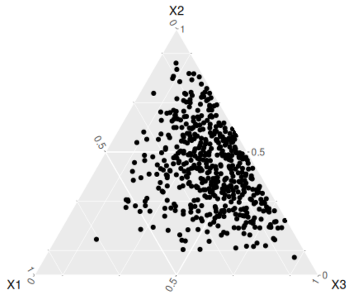
In order to deal with more than 2 components mixture we need to change our model to take into account a *Categorical* latent variable, so we rewrite our model as:

$$Y_i \mid Z_i = z_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})$$
$$Z_1, \dots, Z_n \overset{\text{i.i.d.}}{\sim} \text{Categorical}(p)$$
$$p \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_H)$$
$$\mu_1, \dots, \mu_H \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_{0,i}, 1)$$
$$\frac{1}{\sigma_1^2}, \dots, \frac{1}{\sigma_H^2} \overset{\text{i.i.d.}}{\sim} \text{Gamma}(3, 3)$$

Before talking about how important it is to choose the correct parameters, we define how the Dirichlet distribution changes depending on the value of α and how does this effect the output of our sampling. We use the so called ternary plot to visualize some 3D artificially generated data.



(a) Dirichlet using $\alpha_1 = \alpha_2 = \alpha_3 = 1$



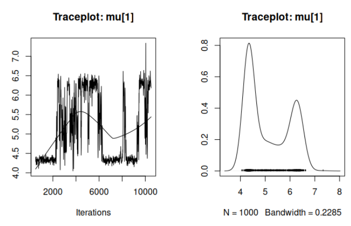
(b) $\alpha_1 = 1.5$, $\alpha_2 = 4$, $\alpha_3 = 3.5$

Figure 9: Dirichlet distributions

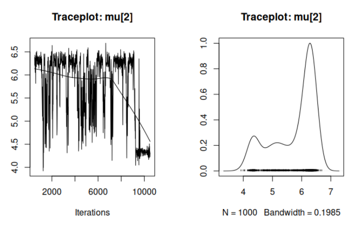
As we can see in Fig. 9 the choice of parameters is fundamental because it directly influences the probabilities of sampling from one of the three components; in particular when giving a completely symmetric prior like in 9a the probabilities are equally spread and as we will see later, whenever we suspect the proportion of data in our clusters is not equally distributed this can cause a lot of label switching in the sampling.
While in 9b we can already see how the data spread more towards the X_2, X_3 components being more representative of cases where clusters don't contain the same amount of data, as we would expect in the 3 components mixture for this dataset.

From what previously said we would expect that this multi component model (for this particular dataset) is really prone to **label switching**, since two components will certainly interfere with each other.
This aspect made us choose our parameters for the mean such that $\mu_{0,1} \neq \mu_{0,2} \neq \mu_{0,3}$ and well separated, otherwise a lot of samples would fall in the same component causing label switching; moreover we also need to be quite careful when setting Dirichlet parameters α_i because they directly affect the probability of being into one of the clusters, and since as we saw from previous sections we strongly believe that most of the observations fall inside only 2 components, we might want to adjust α accordingly.
We will now show an example of traceplots for the posterior means μ_1, μ_2, μ_3 and p_1, p_2, p_3 without caring about label switching to see how impactful it is on the final result.

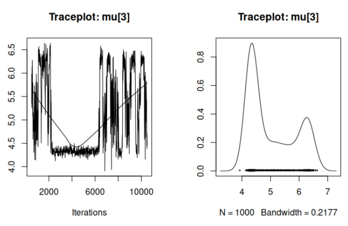
$$\mu_1, \mu_2, \mu_3 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(5, 1), \qquad p \sim \text{Dirichlet}(1, 1, 1)$$



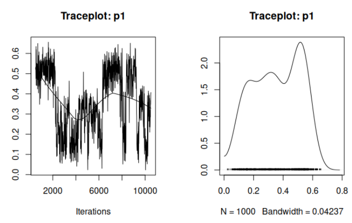
(a) μ_1



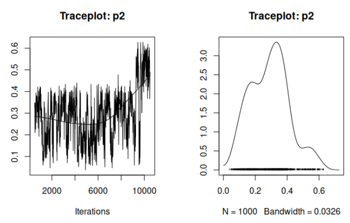
(b) μ_2



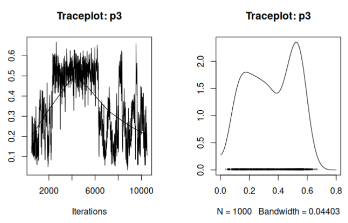
(c) μ_3



(a) p_1



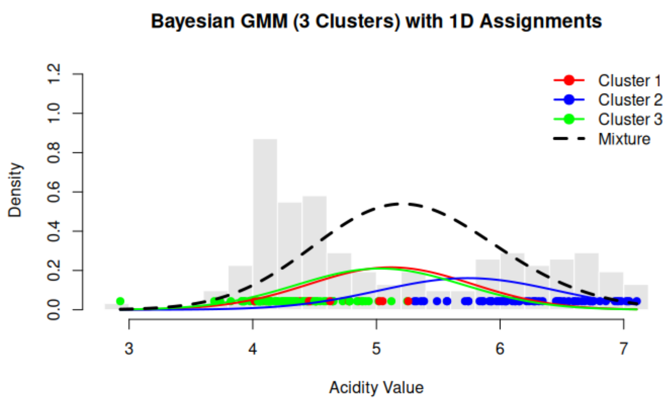
(b) p_2



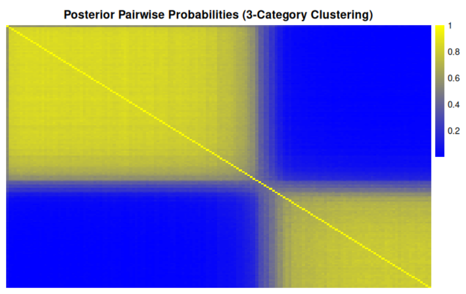
(c) p_3

Figure 10: Traceplots and posteriors probabilities of μ and p with label switching

As expected, the behavior of the posteriors is completely off. Differently from the traceplots observed in the Bernoulli model, we clearly notice high and irregular jumps in the sampling. This indicates that the chain is oscillating between modes of different components, which it ideally shouldn't be able to do. This behavior reflects label switching.



(a) Final Plot of densities and cluster assignments



(b) Heatmap of the Posterior Pairwise Probabilities

Figure 11: Cluster Assignment with wrong parameter setting

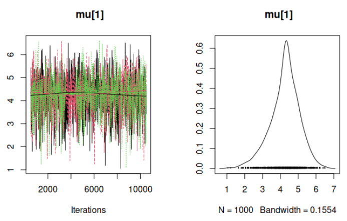
This 11 is how the final cluster assignment looks like. We can clearly see a non-sense mixture paired with a PSM which shows a high uncertainty in observation assignments to a cluster, compared to the one showed in the 2 components in the previous section.

No Label Switching

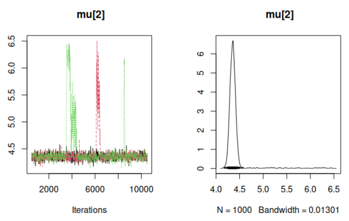
After going through the label switching problem we carefully changed our parameters following what we said before in order to achieve a more coherent result.

$$\mu_{0,1} = 3.5, \quad \mu_{0,2} = 5, \quad \mu_{0,3} = 6.5, \qquad p \sim \text{Dirichlet}(1.5, 4, 3.5)$$

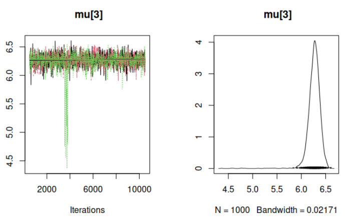
We then run the usual 3 chains *JAGS* sampler and achieved the following posteriors:



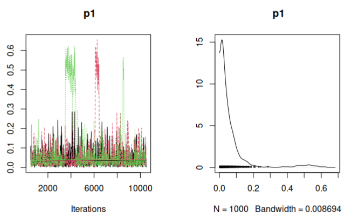
(a) μ_1



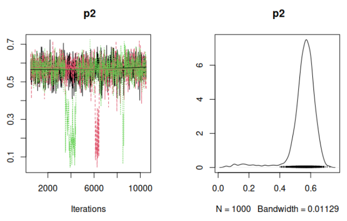
(b) μ_2



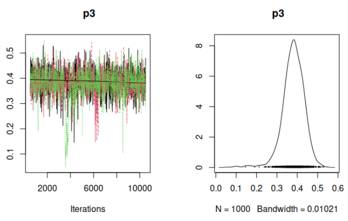
(c) μ_3



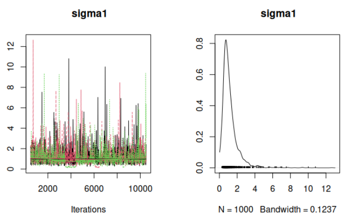
(a) p_1



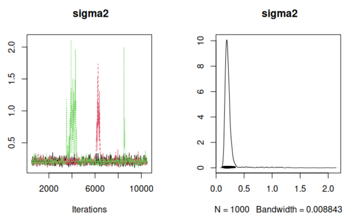
(b) p_2



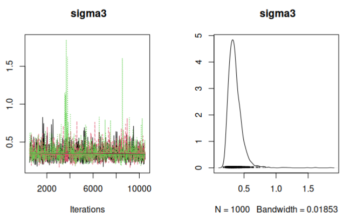
(c) p_3



(a) σ_1^2



(b)σ_2^2

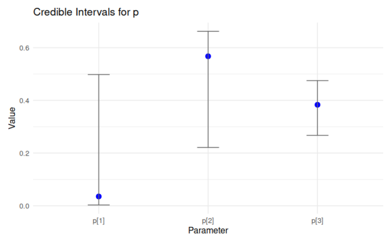


(c)σ_3^2

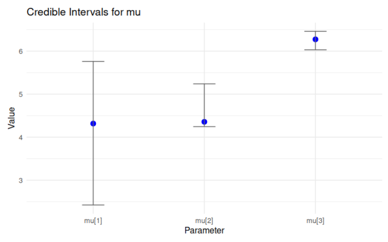
Figure 12: Traceplots and posteriors probabilities of μ and p and σ^2 with label switching considerations

In this case 12 we still see some label switching in a few traceplots but the overall behaviour looks stable and the sudden jumps don't affect the final posterior density.

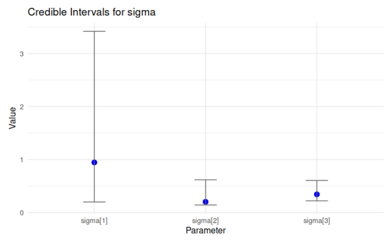
Here there are also the credible intervals for each parameter:



(a) CI p



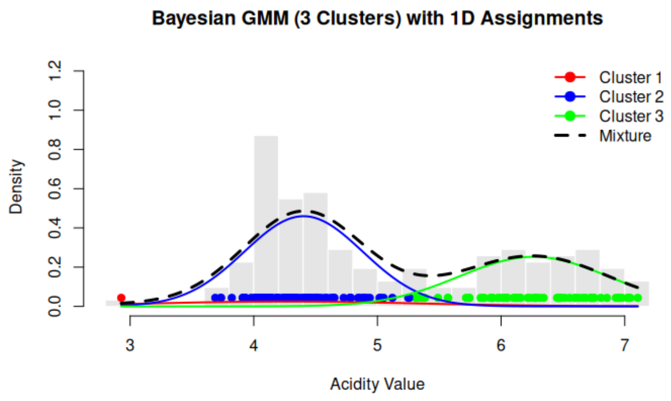
(b) CI μ



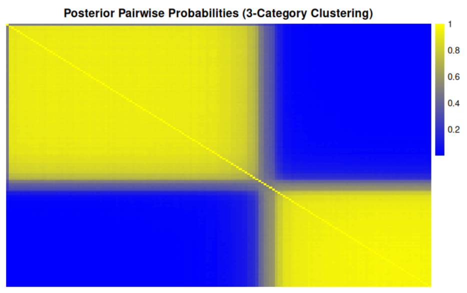
(c) CI σ^2

Figure 13: Credible Intervals at 95% Clearly showing high uncertainty in the first component's distributions of parameters

Finally we can now see how this translates in cluster assignment and PSM:



(a) Final Plot of densities and cluster assignments



(b) Heatmap of the Posterior Pairwise Probabilities

Figure 14: Cluster Assignment with right parameter setting

What we can clearly see in 14 is how we just added a really weak component which take in account for a point which actually just looks like an outlier, the remaining clustering assignments reflects the same behavior described in the Bernoulli suggesting another time that the right number of components is 2. We can also notice how the PSM looks much brighter than the one in Fig 14b showing a much lower uncertainty similar to Fig 11b.

3 Conclusion

In this project, we explored various models for clustering a given dataset using Bayesian methods. The final results provide strong evidence that a 2-component mixture model offers the best fit for the data.

Throughout the analysis, we observed the critical importance of careful parameter initialization, indeed arbitrarily chosen starting values lead to issues such as label switching, which can render the results ambiguous and incorrect.

Unlike traditional clustering techniques such as K-Means, Bayesian approaches not only assign data points to clusters but also gives back full posterior distributions over model parameters. This additional layer of information allows us to quantify uncertainty and gain deeper insights into the structure and reliability of the clustering results.

Overall, the project demonstrates the value of Bayesian inference as a powerful and much more interpretable framework for unsupervised learning.